



Certificación PCAP

Lección 5: Programación Orientada a Objetos (POO)

Actividad 1

ACTIVIDAD LECCIÓN 5

Objetivos

- | Familiarizarse con el uso de las clases y objetos en Python y sus características

Contenido correspondiente a lección 5:

- Programación orientada a objetos en Python
- Herencia

Actividad relacionada con la lección 5:

El objetivo de esta actividad es crear un pequeño código para gestionar el préstamo y devolución de los libros y revistas de una biblioteca. Los libros de la biblioteca tienen un código interno, el título del libro y un código ISBN, mientras que las revistas tienen un código interno, el título de la revista y el número de la revista. Además, este código debe permitir el préstamo y la devolución de los libros y revistas de la biblioteca.

Este código debe poder ser utilizado por el siguiente fragmento de código:

```
libro1 = Libro("L1_123", "La Bestia", "9788408249849")
print("Mi primer libro es:" , libro1)
print("Número ejemplares totales:", libro1.num_totales)
print("Número libros totales:", libro1.num_libros_totales)
print("Número ejemplares prestados:", libro1.num_prestados, "\n")

libro2 = Libro("L2_345", "Últimos días en Berlín", "9788408249856")
print("Mi segundo libro es:" , libro2)
print("Número ejemplares totales:", libro1.num_totales)
print("Número libros totales:", libro1.num_libros_totales)
print("Número ejemplares prestados:", libro1.num_prestados, "\n")

revista1 = Revista("R1_JDJ", "National Geographic", "5")
print("Mi primera revista es:" , revista1)
print("Número ejemplares totales:", revista1.num_totales)
print("Número revistas totales:", revista1.num_revistas_totales)
print("Número ejemplares prestados:", revista1.num_prestados, "\n")

revista2 = Revista("R2_ADA", "National Geographic", "23")
print("Mi segunda revista es:" , revista2)
print("Número ejemplares totales:", revista1.num_totales)
print("Número revistas totales:", revista1.num_revistas_totales)
print("Número ejemplares prestados:", revista1.num_prestados, "\n")

revista1.prestar()
libro1.prestar()
print("Número ejemplares prestados:", libro1.num_prestados)
libro1.devolver()
print("Número ejemplares prestados:", libro1.num_prestados, "\n")
```

Cuya salida por pantalla será:

```
Mi primer libro es: L1_123 - La Bestia - NO PRESTADO (9788408249849)
Número ejemplares totales: 1
Número libros totales: 1
Número ejemplares prestados: 0

Mi segundo libro es: L2_345 - Últimos días en Berlín - NO PRESTADO (9788408249856)
Número ejemplares totales: 2
Número libros totales: 2
Número ejemplares prestados: 0

Mi primera revista es: R1_JDJ - National Geographic - NO PRESTADO (5)
Número ejemplares totales: 3
Número revistas totales: 1
Número ejemplares prestados: 0

Mi segunda revista es: R2_ADA - National Geographic - NO PRESTADO (23)
Número ejemplares totales: 4
Número revistas totales: 2
Número ejemplares prestados: 0

R1_JDJ  PRESTADO
L1_123  PRESTADO
Número ejemplares prestados: 2
L1_123  DEVUELTO
Número ejemplares prestados: 1
```

Indicar que en esta actividad no sólo se va a evaluar que el código funcione y sea correcto, sino también la aplicación de los diferentes conceptos vistos en esta lección.

TIP: Python define una serie de métodos mágicos como, por ejemplo, el método `__str__`, que se pueden sobrescribir en tus clases para obtener el resultado buscado. Concretamente el método `__str__` devuelve una cadena de texto que representa el objeto:

https://docs.python.org/3/reference/datamodel.html?highlight=__str__#object.__str__

Actividad 2

ACTIVIDAD LECCIÓN 5

Objetivos

- | Profundizar en los conceptos vistos en esta lección

Contenido correspondiente a lección 5:

- Programación orientada a objetos en Python
- Herencia

Actividad relacionada con la lección 5:

Las preguntas sobre clases y herencia son muy comunes en el examen de certificación PCAP, por lo que con esta actividad se pretende trabajar en los diferentes conceptos vistos en esta lección. En esta actividad debes enviar para cada pregunta la opción u opciones correctas, así como una breve explicación de tu elección. Es importante que no te olvides de enviar la breve explicación de tu elección ya que si no la envías no se te evaluará la pregunta.

Pregunta 1:

¿Cuál de las siguientes afirmaciones es correcta? (1 opción)

```
class ClaseA:
    A = 1
    def __init__(self, a):
        self.a = a

class ClaseB(ClaseA):
    B = 2
    def __init__(self, a, b):
        ClaseA.A = a
        ClaseB.B = b

o1 = ClaseA(3)
o2 = ClaseB(2, 4)
print(o1.A, end=" ")
print(o1.a, end=" ")
print(o2.B, end=" ")
print(o2.A, end=" ")
print(o2.a, end=" ")
```

- A) Al ejecutarse el código mostrará 1 3 4 2 2
- B) Al ejecutarse el código mostrará 2 3 4 2 2
- C) Al ejecutarse el código mostrará 1 3 4 2 y dará un error
- D) Al ejecutarse el código mostrará 2 3 4 2 y dará un error

Pregunta 2:

Para el siguiente fragmento de código, seleccione la afirmación u afirmaciones correctas (una o varias opciones)

```
class A:
    def __init__(self):
        self.a = 10
    def metodo1(self):
        self.b = 20

o1 = A()
o1.c = 30
o2 = A()
o2.a = 15
o2.metodo1()
```

- A) Si ejecuto `print(o1.a, o1.b, o1.c)` mostrará 10, 20, 30
- B) Si ejecuto `print(o1.a, o1.b, o1.c)` mostrará 15, 20, 30
- C) Si ejecuto `print(o1.a, o1.b, o1.c)` devolverá un error
- D) Si ejecuto `print(o1.a, o1.b)` mostrará 10, 20
- E) Si ejecuto `print(o1.a, o1.b)` mostrará 15, 20
- F) Si ejecuto `print(o1.a, o1.b)` devolverá un error
- G) Si ejecuto `print(o2.a, o2.b, o2.c)` mostrará 15, 20, 30
- H) Si ejecuto `print(o2.a, o2.b, o2.c)` devolverá un error
- I) Si ejecuto `print(o2.a, o2.b)` mostrará 15, 20
- J) Si ejecuto `print(o2.a, o2.b)` devolverá un error
- K) Ninguna de las anteriores es cierta

Pregunta 3:

Si se define una superclase llamada A y una subclase llamada B, cuál o cuáles de los siguientes fragmentos de código debería sustituir al comentario para que cualquier objeto de clase B tuviese el atributo 'a' (una o varias opciones)

```
class A:
    def __init__(self):
        self.a = 1

class B(A):
    def __init__(self):
        # Fragmento de código elegido
        self.b = 2

o = B()
print(o.a)
```

- A) `__init__()`
- B) `A.__init__(self)`
- C) `A.__init__()`
- D) `super.__init__()`
- E) `super.__init(self)`
- F) `super().__init__()`
- G) `super().__init(self)`
- H) No hace falta poner nada

Pregunta 4:

¿Qué devolverá por pantalla el siguiente fragmento de código? (1 opción)

```
class A:
    def metodo(self):
        print('a')

class B:
    def metodo(self):
        print('b')

class C(B):
    def metodo(self):
        print('c')

class D(C,A):
    pass

o= D()
print(issubclass(D,C), ' ', end='')
o.metodo()
```

- A) True a
- B) True b
- C) True c
- D) False a
- E) False b
- F) False c
- G) El programa dará un error.